Dimensionsreduktion

Vorlesung 3, Maschinelles Lernen

Dozenten: Prof. Dr. M. O. Franz, Prof. Dr. O. Dürr

HTWG Konstanz, Fakultät für Informatik

Übersicht

Übersicht

Dimensionsreduktion

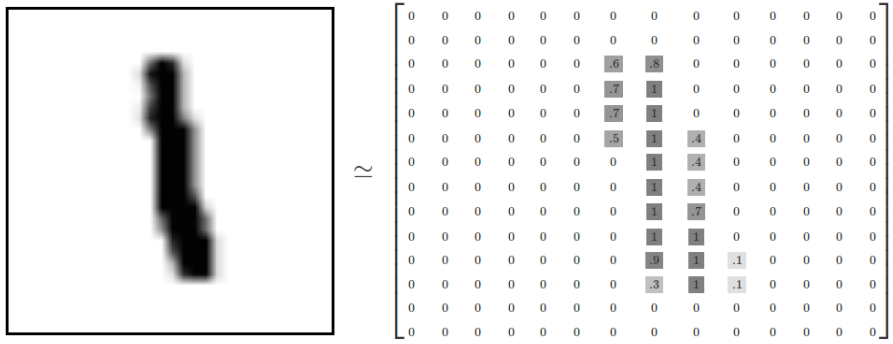
Reale Daten sind häufig sehr hochdimensional (z.B. Genomsequenzen, Bilder, etc.). Der Einsatz von Lernalgorithmen wird dadurch deutlich erschwert, außerdem können hochdimensionale Daten nicht visualisiert werden. Man versucht daher, die Anzahl der Variablen im Lernproblem zu reduzieren (**Dimensionsreduktion**).

Es gibt 2 grundsätzliche Vorgehensweisen:

- **Merkmalsselektion:** es wird eine Untergruppe aus der Gesamtheit aller Variablen anhand bestimmter Kriterien ausgewählt, z.B. Rekonstruktionsfehler, Informationszuwachs etc.
- **Merkmalsextraktion:** es werden wenige neue Variablen (**Merkmale**) aus der Gesamtheit berechnet, die die für das Lernproblem relevante Information möglichst gut repräsentieren.

Beispiel für hochdimensionale Daten: MNIST

MNIST Handwritten Digits



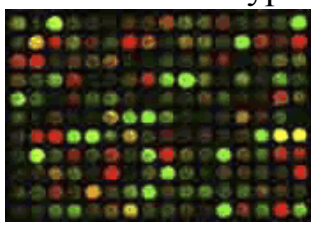
28*28 = 784 features

Row Names	Digit	Pixel_1	Pixel_2	Pixel_3	Pixel_4	Pixel_5		Pixel_256
Sample16	0	0	0	0	0	1		0
Sample78	8	0	1	1	1	1		0
Sample79	3	1	1	1	1	1	...	0
Sample80	2	0	0	0	1	1		1
Sample81	1	0	0	0	0	0		0
Sample82	2	0	0	0	0	1		1
Sample83	4	0	0	0	0	1		1

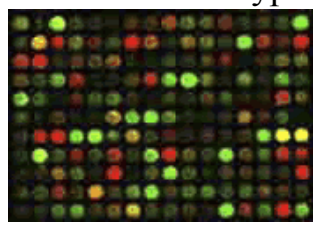
60'000 cases

Beispiel für hochdimensionale Daten: Genexpressionsmuster in Microarrays

Breast Cancer Type I



Breast Cancer Type II



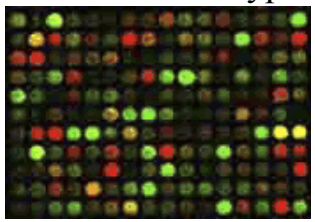
~30'000 Features

~100 Chips Patients

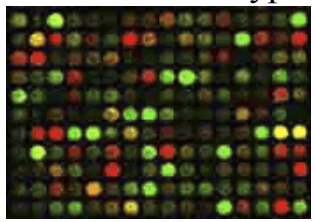
Patient	Cancer Type	Gene 1	Gene 2	Gene 3		Gene 30000
1	Cancer I	3.5	1.4	0.2		54
2	Cancer II	3	1.4	0.2	...	3.3
3	Cancer I	3.2	1.6	0.5	...	45
4	Cancer II	3.5	1.4	0.2	...	44
...	...	...	...	...	...	...
100	Cancer II	3	1.4	0.2		65.0

Beispiel für hochdimensionale Daten: Genexpressionsmuster in Microarrays

Breast Cancer Type I



Breast Cancer Type II



~30'000 Features

~100 Chips Patients

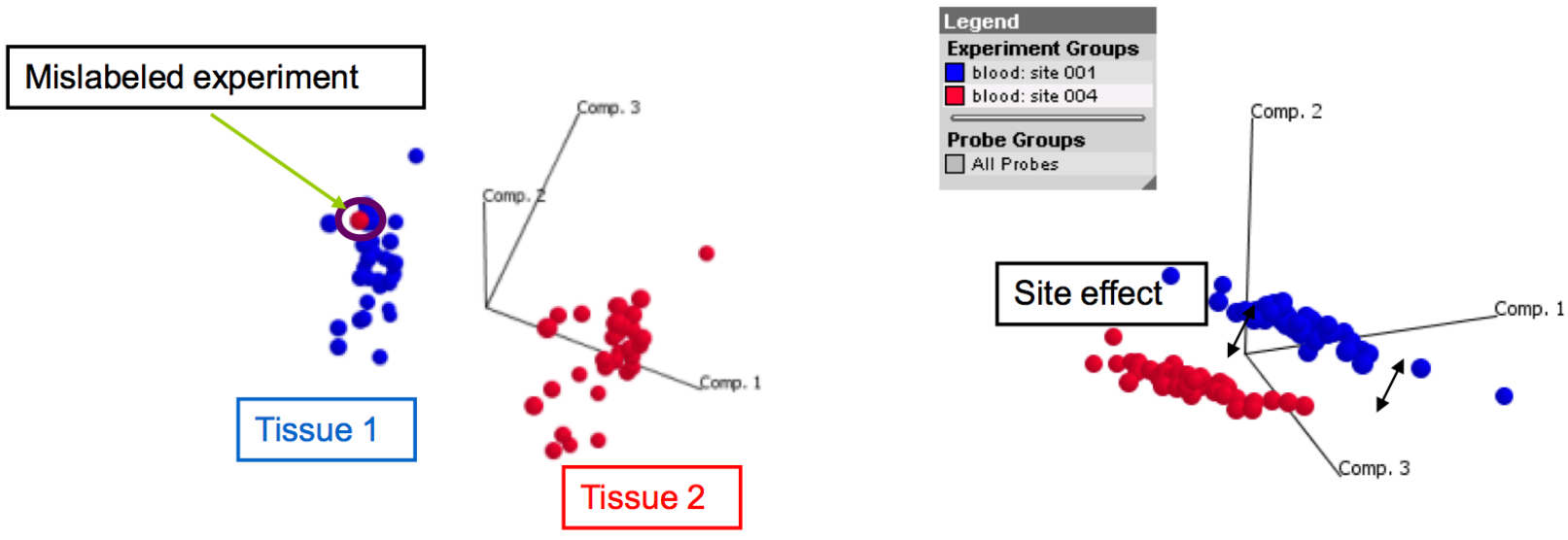
Patient	Cancer Type	Gene 1	Gene 2	Gene 3		Gene 30000
1	Cancer I	3.5	1.4	0.2		54
2	Cancer II	3	1.4	0.2	...	3.3
3	Cancer I	3.2	1.6	0.5	...	45
4	Cancer II	3.5	1.4	0.2	...	44
...	...	...	...	...	...	...
100	Cancer II	3	1.4	0.2		65.0

Wie findet man Fehler in 30.000 Dimensionen?

		Var1	Var2	Var3	Var4		Var29998	Var29999	Var30000
2	Exp 1	0.72575697	0.90912727	0.34183219	0.05807989		0.89604396	0.82978394	0.67709992
3	Exp 2	0.86738472	0.18892562	0.10061359	0.86472149		0.42230826	0.29717662	0.84183402
4	Exp 3	0.536584	0.1432163	0.48031828	0.54135801		0.35761078	0.93715841	0.09946435
5	Exp 4	0.9750189	0.01024606	0.5091362	0.15267143		0.21621312	0.18110699	0.04165032
6	Exp 5	0.54058652	0.86667423	0.7723371	0.29263193		0.34003667	0.99724025	0.50546296
7	Exp 6	0.51420486	0.68710973	0.53443674	0.39784944		0.09120201	0.02095151	0.7874859
8	Exp 7	0.84066935	0.77631801	0.88487998	0.44679161		0.06190021	0.5841718	0.79347598
9	Exp 8	0.31751833	0.16318624	0.79276299	0.08440983		0.83189181	0.57771774	0.08795787
10	Exp 9	0.25660099	0.26809617	0.87877424	0.41877501		0.23589796	0.84845295	0.12819384
11	Exp 10	0.2133815	0.35005992	0.72679513	0.93096216		0.06484648	0.87936602	0.98808592
12	Exp 11	0.67621687	0.82208455	0.41252689	0.51356375		0.83390677	0.58056355	0.53156471
13	Exp 12	0.83698302	0.39427289	0.64891165	0.38946918		0.86980017	0.02408343	0.1279679
14	Exp 13	0.55158332	0.83563428	0.08601078	0.95342836		0.73062691	0.69545093	0.9684515
15	Exp 14	0.88240173	0.93238978	0.34213598	0.25428121		0.7835426	0.08090319	0.34138029
16	Exp 15	0.35257344	0.04100738	0.08441876	0.54433121		0.39901506	0.75584409	0.90975939
17	Exp 16	0.66411782	0.26894263	0.41570473	0.90789058		0.22048541	0.06748163	0.56058675
18	Exp 17	0.37758333	0.87416083	0.34331478	0.57516207		0.05496016	0.65253258	0.13182802
19	Exp 18	0.16254575	0.45823383	0.41947507	0.20725022		0.76805359	0.29279849	0.17038373
20	Exp 19	0.09717491	0.16687942	0.69490897	0.00982516	...	0.91838419	0.33659043	0.92492301
21	Exp 20	0.18632352	0.26820246	0.43650327	0.87902106		0.24433306	0.63146315	0.32357043
22	Exp 21	0.16356459	0.64175502	0.54539885	0.44821048		0.69554721	0.23790817	0.16114107
23	Exp 22	0.50602018	0.20370984	0.38225406	0.797264		0.14490904	0.44722882	0.31422467
24	Exp 23	0.30294307	0.73866033	0.78865558	0.3532843		0.04564231	0.00854157	0.93346583
25	Exp 24	0.37893552	0.36223255	0.56103558	0.68409418		0.37509117	0.65625123	0.85521739
26	Exp 25	0.48307778	0.70717319	0.73891708	0.56796612		0.21659263	0.89437545	0.3689527
27	Exp 26	0.76200969	0.71219127	0.01349004	0.3164314		0.13000069	0.06598902	0.64582494
28	Exp 27	0.73420576	0.48717756	0.90633582	0.78943633		0.39879527	0.66474155	0.87347295
29	Exp 28	0.81817237	0.96946477	0.10527094	0.34758947		0.41245137	0.05720508	0.64870324
30	Exp 29	0.46189251	0.92516654	0.88555359	0.94335229		0.5822599	0.23685582	0.45742172
31	Exp 30	0.64594505	0.89508066	0.00344762	0.01764184		0.98594893	0.73566371	0.65856274
32	Exp 31	0.29977128	0.57625009	0.42203689	0.53401962		0.73209191	0.78395094	0.79902787
33	Exp 32	0.24302476	0.30748217	0.13336479	0.06307744		0.43565341	0.80502196	0.54948119
34	Exp 33	0.10313471	0.81037692	0.1019815	0.95377739		0.93837378	0.32174496	0.06968822
35	Exp 34	0.57267352	0.90832149	0.01010143	0.79129947		0.99566332	0.60311529	0.94254175
36	Exp 35	0.50726326	0.37511111	0.42905886	0.07225932		0.04399464	0.50687479	0.55638528
37	Exp 36	0.25945485	0.87641759	0.21325063	0.07747172		0.76669305	0.62018391	0.6870877
38	Exp 37	0.74949084	0.4022447	0.67541199	0.63615423		0.10450498	0.70701564	0.5133567
39	Exp 38	0.02851173	0.72048367	0.64104403	0.85594184		0.6012205	0.68162603	0.2020788
40	Exp 39	0.34281377	0.00438227	0.72128967	0.95437483		0.23733739	0.90221594	0.82944182
41	Exp 40	0.59605972	0.47063129	0.0386638	0.01654464		0.75888817	0.71401908	0.68048745
42	Exp 41	0.1990435	0.56673445	0.81536695	0.63442106		0.07796896	0.14863039	0.70065248
43	Exp 42	0.08196709	0.21791967	0.05331609	0.32315459		0.35220877	0.04819191	0.77405895
44	Exp 43	0.40558134	0.79050103	0.27871425	0.24711674		0.66015164	0.00739487	0.64399704
45	Exp 44	0.4931796	0.76353204	0.99650294	0.27611642		0.06225584	0.02230363	0.00332453
46	Exp 45	0.82594597	0.0448367	0.46463296	0.75938489		0.82920136	0.36006835	0.11734431

Visualisierung durch Dimensionsreduktion

Beispiel: nur die zwei ”wichtigsten” Dimensionen werden visualisiert. ”Wichtig” heißt hier: Dimensionen, in denen die Daten möglichst breit streuen (d.h. hohe Varianz haben), so dass ihre internen Strukturen sichtbar werden.



Wiederholung: eindimensionale Korrelation und Kovarianz

Korrelation zweier Variablen x und y:

$$C_{xy} = \mathcal{E}[xy] = \sum_{x_i \in \mathcal{X}} \sum_{y_j \in \mathcal{Y}} xy\, p(x, y)$$

- Wenn x und y voneinander statistisch abhängig sind, dann sind sie oft gleichzeitig positiv oder gleichzeitig negativ (sie **kovariieren**). Somit ist ihr Produkt oft groß und damit auch ihre Korrelation.
- Die Korrelation ist umso höher, je größer die Mittelwerte der Variablen sind, unabhängig davon, ob sie zusätzlich kovariieren oder nicht.
- Ein besseres Maß für die statistische Abhängigkeit ist daher die **Kovarianz** von x und y:

$$\sigma_{xy} = \mathcal{E}[(x - \mu_x)(y - \mu_y)] = \sum_{x_i \in \mathcal{X}} \sum_{y_j \in \mathcal{Y}} (x - \mu_x)(y - \mu_y)\, p(x, y).$$

Kovarianzmatrix

Die **Kovarianzmatrix** Σ besteht aus den paarweisen Kovarianzen σ_{ij} jeder Variablen x_i mit allen anderen x_j (incl. x_i selbst)

$$\Sigma = (\sigma_{ij}) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots \\ \sigma_{21} & \sigma_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots \\ \sigma_{21} & \sigma_2^2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Die Kovarianzmatrix ist *symmetrisch*, d.h. es gilt $\Sigma_{ij} = \Sigma_{ji}$. Die Diagonalelemente $\sigma_{ii} = \sigma_i^2$ sind die Varianzen der einzelnen x_i.

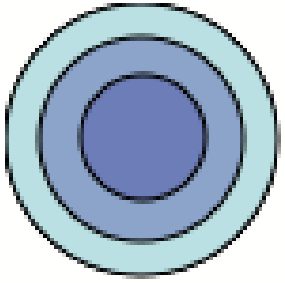
Ordnen wir die Variablen x_i in Merkmalsvektoren $\mathbf{x}$ mit Mittelwert $\boldsymbol{\mu}_\mathbf{x}$, so lässt sich Σ als Erwartungswert des **äußeren Produkts** berechnen:

$$\Sigma_{ij} = \mathcal{E}[(x_i - \mu_i)(x_j - \mu_j)] \quad \Rightarrow \quad \Sigma = \mathcal{E}[(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})(\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top]$$

Σ beschreibt die Form und Orientierung der Daten im Merkmalsraum.

Demo: https://www.jgoertler.com/visual-exploration-gaussian-processes/

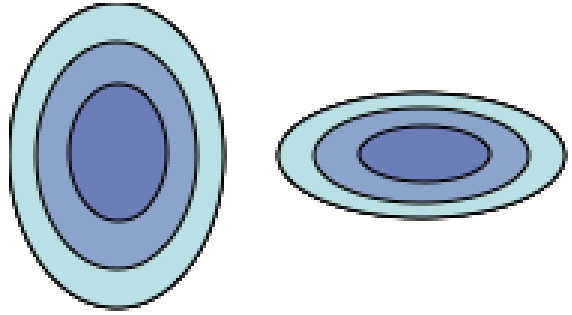
2D-Beispiele für Kovarianzmatrizen



$$\begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

Fall 1: Diagonalelemente gleich, nichtdiagonale Elemente sind 0.

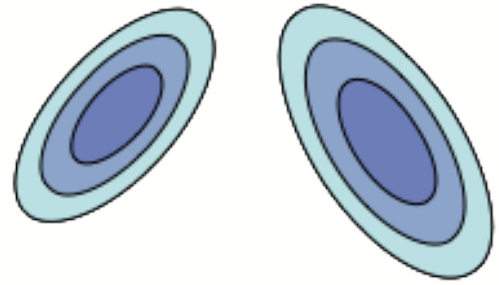
- Varianz ist in alle Richtungen gleich.
- Daten sind isotrop verteilt.
- beide Variablen sind unkorreliert.



$$\begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}$$

Fall 2: Diagonalelemente ungleich, nichtdiagonale Elemente sind 0.

- Daten sind in einer Ellipse verteilt, die an den Achsen ausgerichtet ist.
- beide Variablen sind immer noch unkorreliert.



$$\begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{pmatrix}$$

Fall 3: Alle Einträge von Σ sind ungleich 0, $\sigma_{12} = \sigma_{21}$.

- Verteilung der Daten folgt beliebig ausgerichtete Ellipsen.
- beide Variablen sind korreliert.

Varianz entlang einer Raumrichtung

Gesucht ist die Varianz entlang einer Raumrichtung. Eine beliebige Richtung im $\mathbb{R}^d$ wird durch einen Einheitsvektor $\mathbf{q}$ dargestellt, auf den die Datenpunkte $\mathbf{x}$ nach Abzug des Mittelwertes $\boldsymbol{\mu}_\mathbf{x}$ projiziert werden:

$$a = (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \mathbf{q} = \mathbf{q}^\top (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) \quad \text{mit} \quad \|\mathbf{q}\| = \sqrt{\mathbf{q}^\top \mathbf{q}} = 1$$

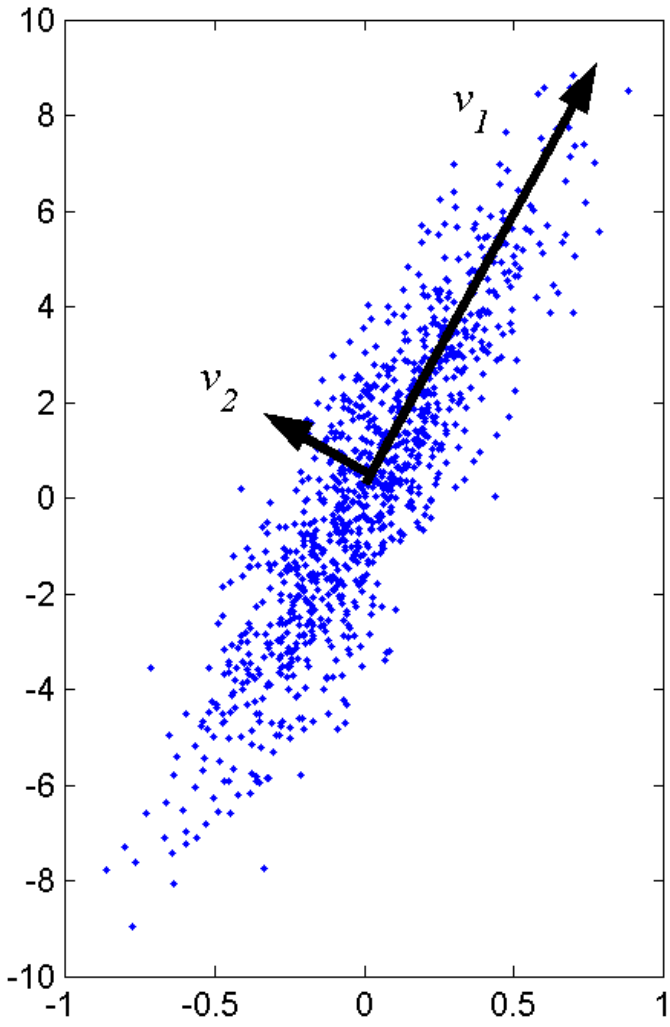
Varianz in Richtung q:

$$\begin{aligned}
\sigma^2(\mathbf{q}) &= \mathcal{E}[a^2] \\
&= \mathcal{E}[(\mathbf{q}^\top (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}))((\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top \mathbf{q})] \\
&= \mathbf{q}^\top \mathcal{E}[((\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}))((\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\top] \, \mathbf{q} \\
&= \mathbf{q}^\top \Sigma \mathbf{q}
\end{aligned}$$

Die Varianzen entlang aller Raumrichtungen haben i.A. jeweils Paare von Maxima und Paare von Minima, die senkrecht aufeinander stehen.

Übersicht

Hauptkomponentenanalyse (PCA)



In der Hauptkomponentenanalyse (engl. Principal Component Analysis, PCA) wird nach Richtungen in den Daten gesucht, entlang derer die Daten extremale (d.h. maximale oder minimale) Varianz haben.

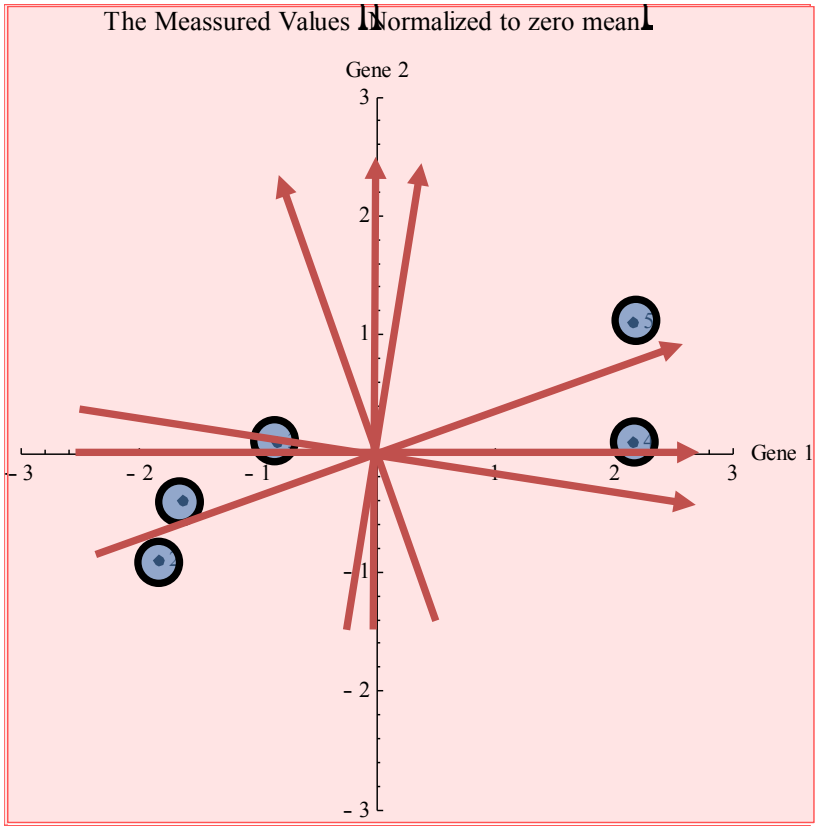
Anwendung: Dimensionsreduktion durch Weglassen der Richtungen mit der kleinsten Varianz.

Annahme: die Daten seien zentriert, d.h. der Schwerpunkt

$$\boldsymbol{\mu}_{\mathbf{x}} = \frac{1}{n} \sum_i \mathbf{x}_i$$

liegt am Ursprung.

Hauptkomponentenanalyse als Rotation



Das Koordinatensystem wird so rotiert, dass die Daten die größte Varianz entlang der ersten Koordinatenachse haben, die zweithöchste Varianz entlang der zweiten Achse usw.

Im Beispiel ist die Varianz maximal bei einer Rotation um ca. -18°. Da die Daten zweidimensional sind, ist damit die Gesamtrotation festgelegt.

PCA als Optimierungsproblem

Wie vorhin gezeigt, ist die Varianz entlang einer Raumrichtung $\mathbf{q}$ durch $\sigma^2(\mathbf{q}) = \mathbf{q}^\top \Sigma \mathbf{q}$ gegeben. Die PCA ist also durch ein **Optimierungsproblem** definiert:

$$\text{maximiere} \quad \sigma^2(\mathbf{q}), \quad \mathbf{q} \in \Omega,$$

oder, äquivalent, über ein Minimierungsproblem (jedes Maximierungsproblem kann durch Umkehrung des Vorzeichens $f(\mathbf{q}) = -\sigma^2(\mathbf{q})$ in ein Minimierungsproblem umgewandelt werden):

$$\text{minimiere} \quad f(\mathbf{q}), \quad \mathbf{q} \in \Omega.$$

$f(\mathbf{q})$ nennt man die Zielfunktion des Optimierungsproblems, Ω den zulässigen Bereich des Problems, d.h. die Menge, die man zur Optimierung durchsuchen muss (in unserem Fall die Menge aller Raumrichtungen).

Optimierung unter Nebenbedingungen

Oft wird der zulässige Bereich implizit durch Gleichheits- und/oder Ungleichheitsbedingungen beschrieben. In einem solchen Fall spricht man von einem

Optimierungsproblem unter Nebenbedingungen

$$\text{minimiere} \quad f(\mathbf{q}), \quad \mathbf{q} \in \Omega$$

unter den Nebenbedingungen:

$$\begin{aligned} g_i(\mathbf{q}) &\leq 0, \quad i = 1 \ldots k \quad \text{(Ungleichheitsbedingungen)} \\ h_i(\mathbf{q}) &= 0, \quad i = 1 \ldots m \quad \text{(Gleichheitsbedingungen)} \end{aligned}$$

Durch die Funktionen $g_i(\mathbf{q})$ und $h_i(\mathbf{q})$ wir die Wahl von $\mathbf{q}$ auf die Bereiche von Ω eingeschränkt, in denen die Gleichheits- und Ungleichheitsbedingungen erfüllt sind.

Welches ist der zulässige Bereich für die PCA?

Lässt man für unser Problem

$$\text{minimiere} \quad -\mathbf{q}^\top \Sigma \mathbf{q}, \quad \mathbf{q} \in \Omega$$

$\Omega = \mathbb{R}^n$ zu, dann gibt es keine eindeutige endliche Lösung. Wir müssen also den zulässigen Bereich durch Nebenbedingungen einschränken. Vorhin wurde festgelegt, dass eine Raumrichtung durch einen Einheitsvektor dargestellt werden soll, d.h. $\|\mathbf{q}\| = \sqrt{\mathbf{q}^\top \mathbf{q}} = 1$ oder einfacher $\mathbf{q}^\top \mathbf{q} = 1$. Somit haben wir eine Gleichheitsbedingung, die überhaupt erst eine endliche Lösung ermöglicht:

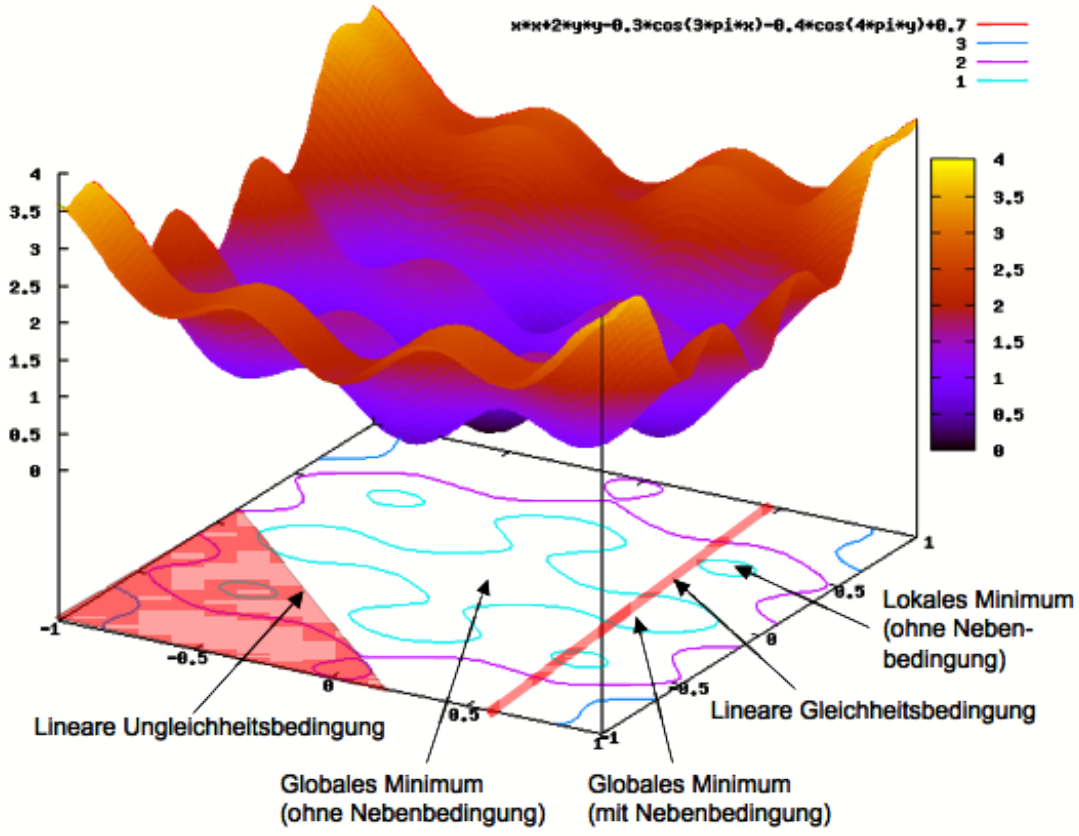
PCA-Optimierungsproblem

$$\text{minimiere} \quad -\mathbf{q}^\top \Sigma \mathbf{q}, \quad \mathbf{q} \in \mathbb{R}^n$$

unter der Nebenbedingung

$$h(\mathbf{q}) = \mathbf{q}^\top \mathbf{q} - 1 = 0.$$

Wiederholung: Minima ohne Nebenbedingungen



Satz (Fermat)

Sei f differenzierbar. Eine notwendige Bedingung für ein Minimum an der Stelle $\mathbf{q}^*$ ist

$$\nabla_{\mathbf{q}} f(\mathbf{q}^*) = 0.$$

Das Minimum läßt sich z.B. durch Lösen der Gleichung $\nabla_{\mathbf{q}} f(\mathbf{q}) = 0$ finden.

Diese Methode funktioniert nicht unter Nebenbedingungen, da der zulässige Bereich eingeschränkt ist.

Wie findet man Minima unter Gleichheitsbedingungen?

Allgemein definiert man die Lagrangefunktion L mit den Lagrange-Multiplikatoren β_i als

$$L(\mathbf{q}, \boldsymbol{\beta}) = f(\mathbf{q}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{q}).$$

Satz (Lagrange)

Notwendige Bedingung für ein Minimum unter den Nebenbedingungen $h_i(\mathbf{q}) = 0$:

1. $\nabla_{\boldsymbol{\beta}} L(\mathbf{q}, \boldsymbol{\beta}) = 0$ (entspricht $h_i(\mathbf{q}) = 0$)
2. $\nabla_{\mathbf{q}} L(\mathbf{q}, \boldsymbol{\beta}) = 0$

Die Optimierung von f unter Nebenbedingungen wird also durch eine Optimierung von $L(\mathbf{q}, \boldsymbol{\beta})$ ohne Nebenbedingungen ersetzt.

Anwendung auf die Hauptkomponentenanalyse

Lagrangefunktion der PCA:

$$L(\mathbf{q}, \beta) = -\mathbf{q}^\top \Sigma \mathbf{q} + \beta(\mathbf{q}^\top \mathbf{q} - 1).$$

Ableitung nach $\mathbf{q}$:

$$\nabla_{\mathbf{q}} L(\mathbf{q}, \beta) = -\Sigma \mathbf{q} + \beta \mathbf{q}$$

Nach dem Satz von Lagrange muss am Minimum

$$\nabla_{\mathbf{q}} L(\mathbf{q}, \beta) = -\Sigma \mathbf{q} + \beta \mathbf{q} = 0$$

sein, d.h. wir erhalten eine Eigenwertgleichung :

$$\boxed{\Sigma \mathbf{q} = \beta \mathbf{q}}$$

Die Eigenvektoren der Kovarianzmatrix sind also die Richtungen extremaler Varianz!

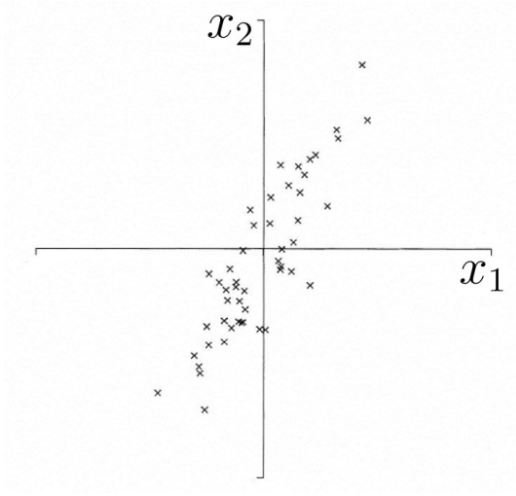
Hauptkomponenten

- Die Kovarianzmatrix ist symmetrisch. Aus der linearen Algebra wissen wir, dass die Eigenwertgleichung dann immer eine Lösung hat und die Eigenvektoren zueinander **orthogonal** sind.
- Insbesondere ist Σ **orthogonal diagonalisierbar**, d.h. es gibt eine orthogonale Transformation Q, so dass

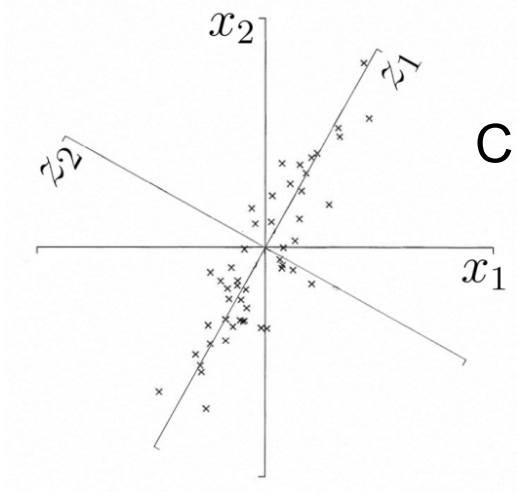
$$\Sigma = Q^\top \Sigma' Q = Q^\top \begin{pmatrix} \beta_1 & 0 & \cdots \\ 0 & \beta_2 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} Q.$$

- Q ist in unserem Fall eine Rotationsmatrix, also genau die vorhin gesuchte PCA-Rotation. Ihre Spalten sind die Eigenvektoren $\mathbf{q}_i$ (normiert auf Betrag 1), die eine neue Orthonormalbasis im $\mathbb{R}^n$ bilden, in der die Kovarianzmatrix Σ' diagonal ist. Diese neuen normierten Basisvektoren heißen Hauptkomponenten.

Neue Variablen: Principal Component Scores



PCA (rotation)



C is diagonal!

Nach der Rotation sind die Koordinatenachsen entlang der Richtungen extremaler Varianz ausgerichtet. Die Projektionen eines Datenpunktes auf die $\mathbf{q}_i$

$$a_i = \mathbf{q}_i^\top \mathbf{x} = \mathbf{x}^\top \mathbf{q}_i$$

sind die neuen Variablenwerte (engl. Scores) in der Hauptkomponentenbasis.

Kovarianzstruktur der neuen Variablen

Varianz der neuen Variablen:

$$\sigma_{i'}^2 = \mathcal{E}[a_i^2] = \sigma^2(\mathbf{q}_i) = \mathbf{q}_i^\top \Sigma' \mathbf{q}_i = \beta_i.$$

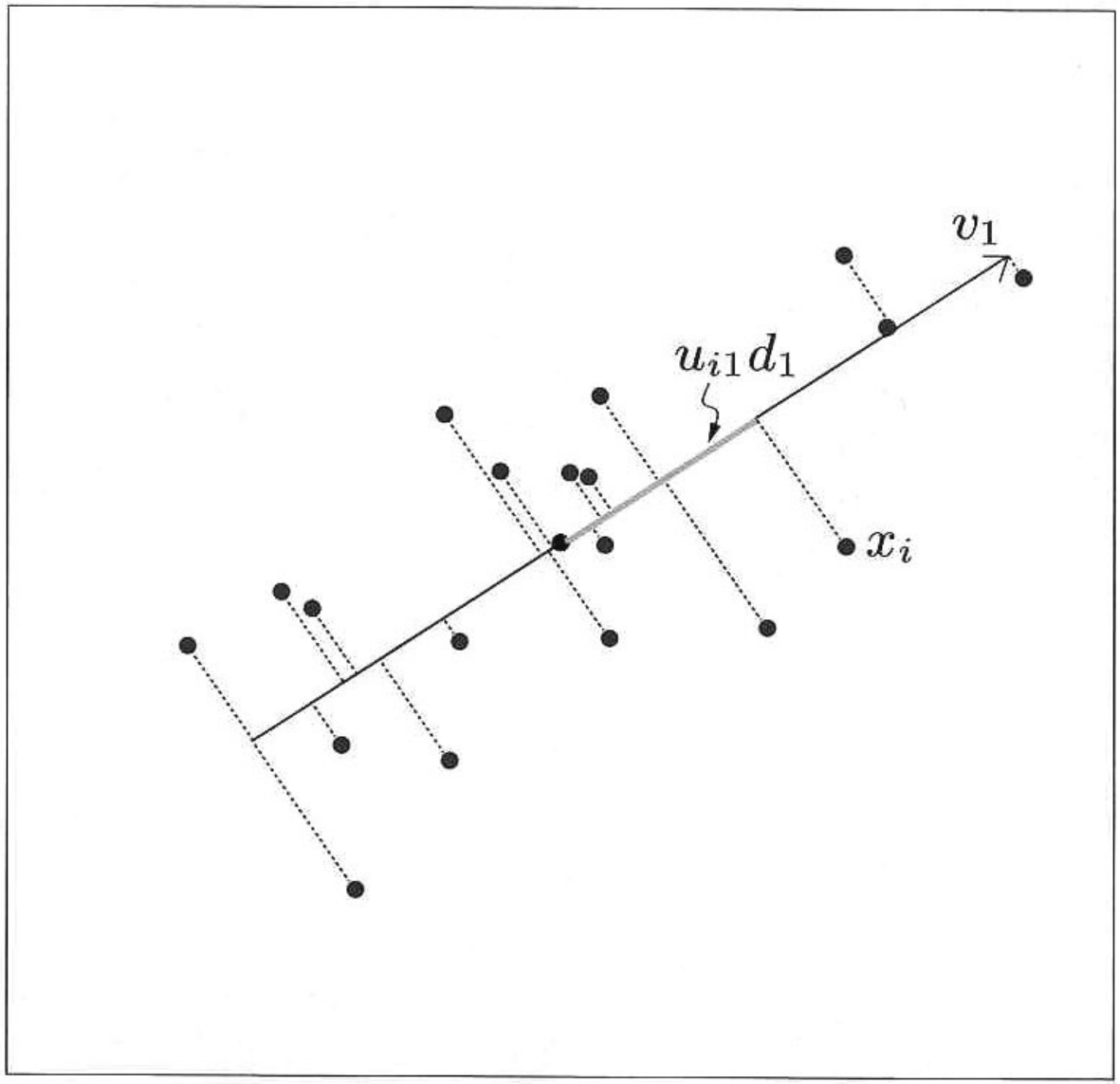
Die Eigenwerte β_i von Σ sind also die Varianzen $\sigma_{i'}^2$ der Scores a_i. Da Varianzen immer größer oder gleich 0 sind, gilt $\beta_i \geq 0$, d.h. Σ ist **positiv semidefinit**.

Die transformierte Kovarianzmatrix Σ' ist diagonal. Somit gilt für die Kovarianz zwischen a_i und a_j

$$\sigma_{i'j'} = \Sigma'_{ij} = 0 \quad \text{für} \quad i \neq j.$$

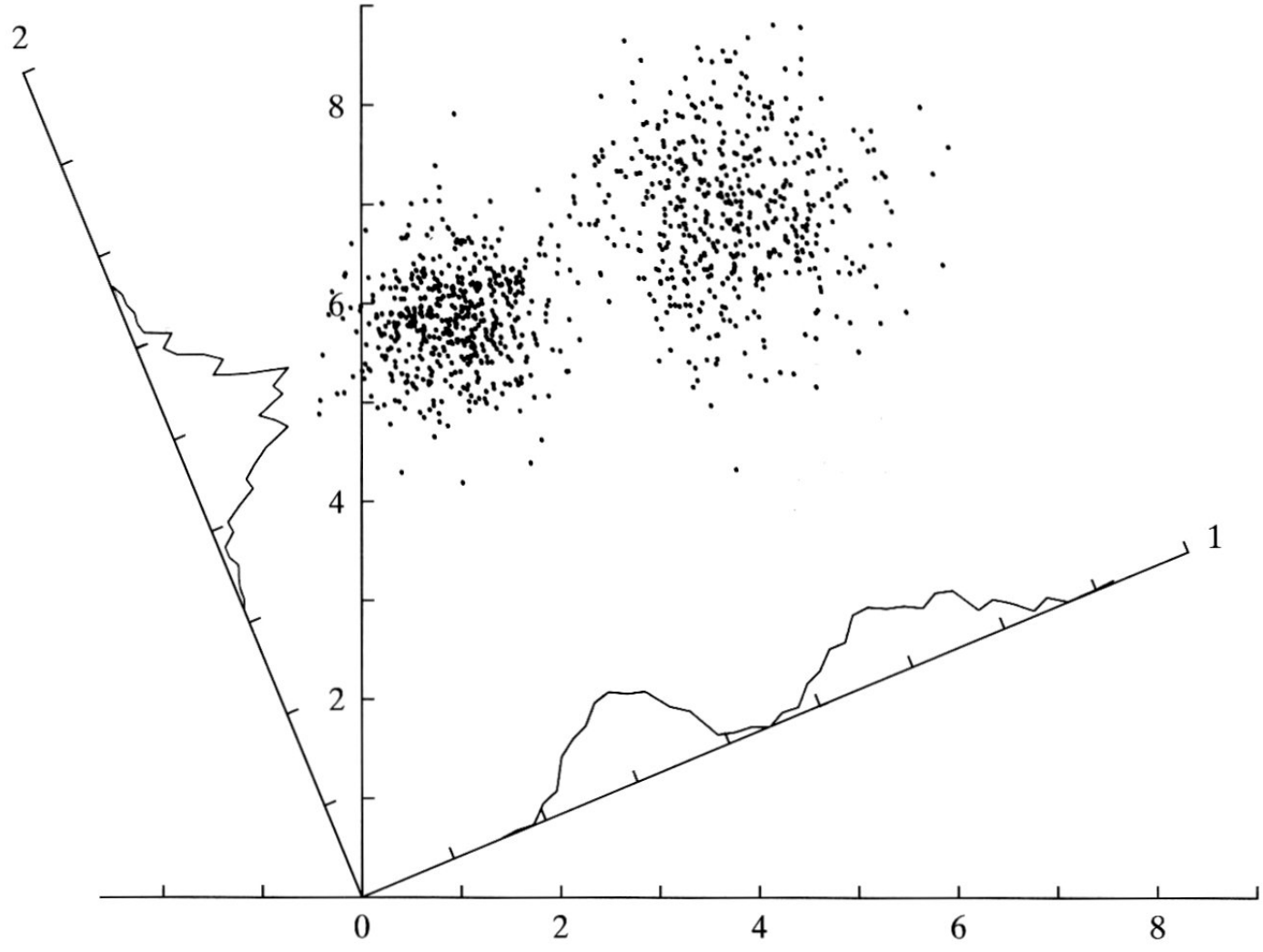
Die neuen Variablen sind somit untereinander **unkorreliert**.

2D-Beispiel: a_1 bzw. Projektion auf 1. Hauptkomponente



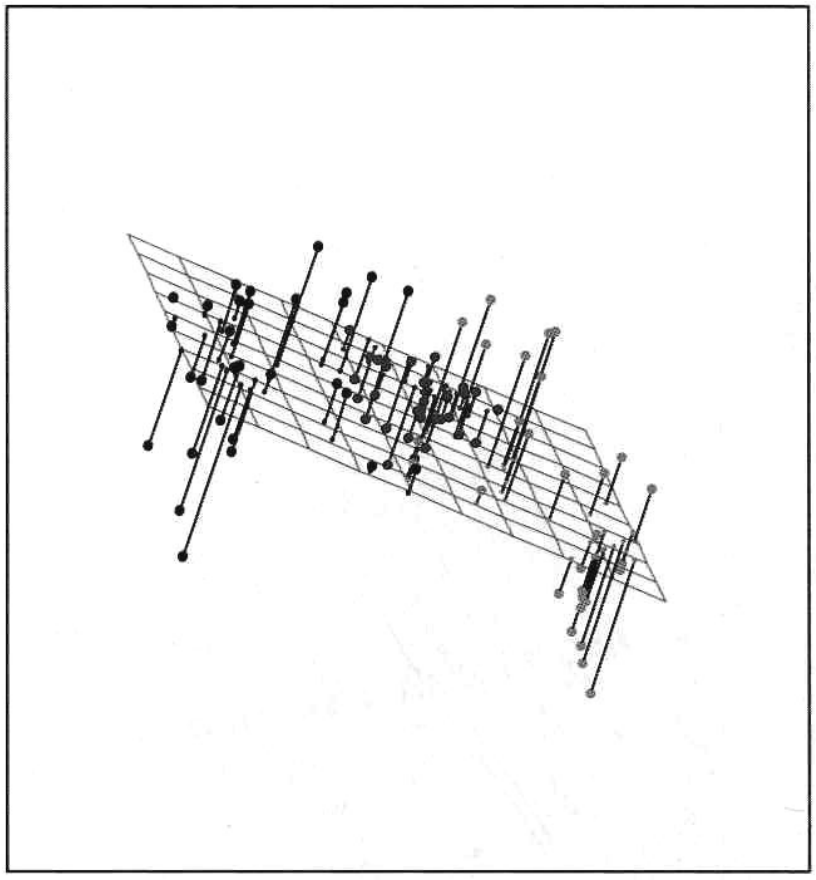
[Hastie et al., 2001]

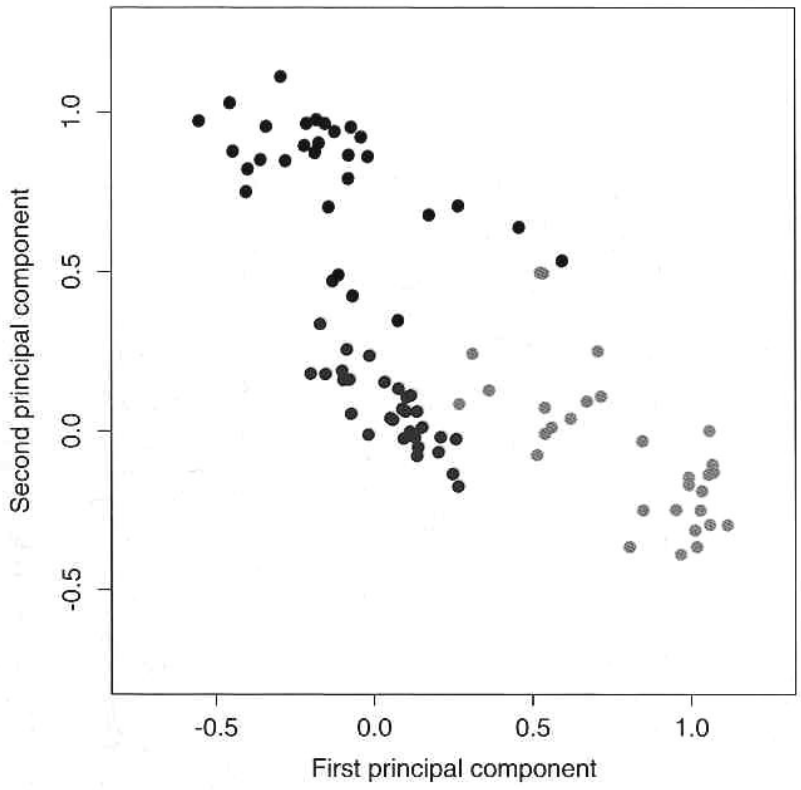
2D-Beispiel: a_1 und a_2



[Haykin 99]

3D-Beispiel: 1. und 2. Hauptkomponente





[Haykin 99]

Übersicht

Dimensionsreduktion mit Hauptkomponenten

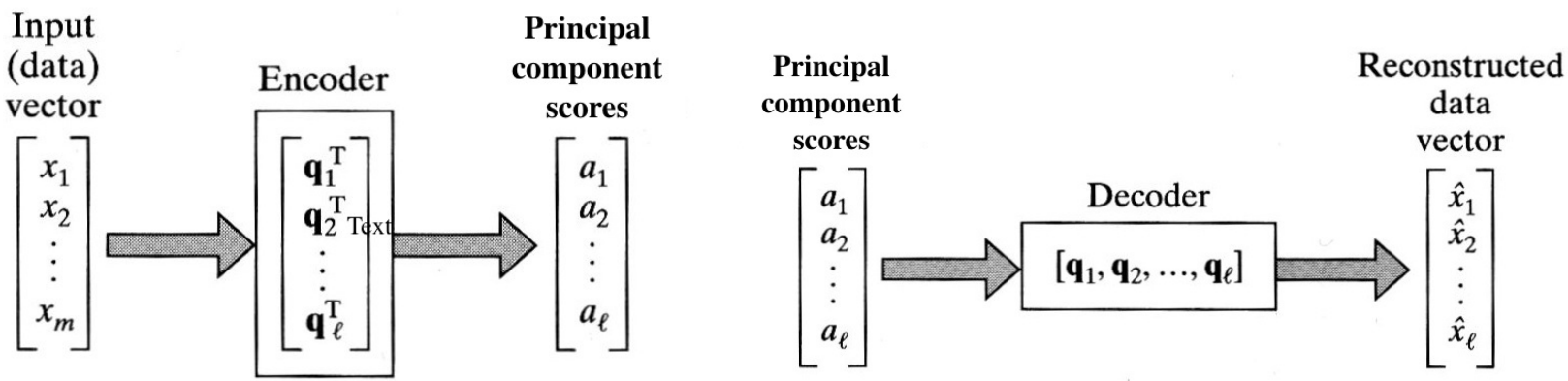
Prinzip: Zur Dimensionsreduktion werden die Hauptkomponenten nach ihrer Varianz absteigend geordnet. Es werden nur die ℓ Hauptkomponenten mit der größten Varianz beibehalten, Hauptkomponenten kleinerer Varianz werden verworfen.

Codierungsschritt: die Daten werden in den Unterraum der ℓ größten Varianzen projiziert

$$\mathbf{x} \quad \longrightarrow \quad a_i = \mathbf{q}_i^\top \mathbf{x}, \;\; i = 1 \ldots \ell$$

Decodierungsschritt: Rekonstruktion aus den Projektionen

$$\hat{\mathbf{x}} = \sum_{i=1}^{\ell} a_i \mathbf{q}_i$$



Die PCA-Rekonstruktion ist optimal

Von allen Orthonormalbasen eines ℓ-dimensionalen Unterraums ist die PCA-Basis optimal im Sinne, dass sie den mittleren quadratischen Rekonstruktionsfehler minimiert. Wenn wir alle Hauptkomponenten behalten, ist die Rekonstruktion exakt:

$$\mathbf{x} = \sum_{i=1}^{n} a_i \mathbf{q}_i,$$

bei $\ell < n$ Komponenten ist sie nur eine Näherung:

$$\hat{\mathbf{x}} = \sum_{i=1}^{\ell} a_i \mathbf{q}_i.$$

Mittlerer quadratischer Fehler:

$$\mathcal{E}[\|\mathbf{x} - \hat{\mathbf{x}}\|^2] = \mathcal{E}[\left\| \sum_{i=\ell+1}^{n} a_i \mathbf{q}_i \right\|^2] = \sum_{i=\ell+1}^{n} \mathcal{E}[a_i^2] = \sum_{i=\ell+1}^{n} \sigma_{i'}^2$$

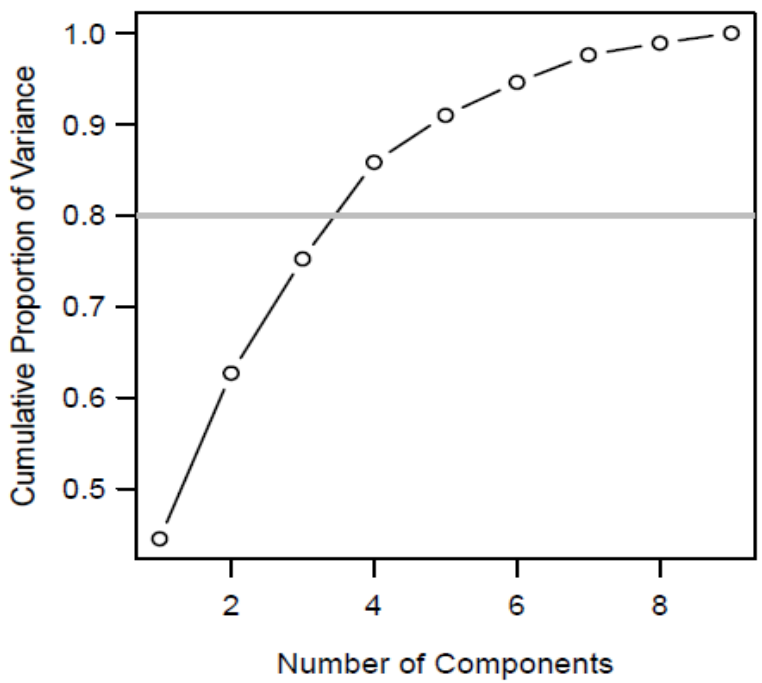
Der Fehler jeder anderen Variablenkombination wäre größer!

Wieviele Hauptkomponenten braucht man?

Gesamtvarianz (ändert sich nicht bei Rotation):

$$\sigma^2_{\text{ges}} = \sum_{i=1}^{n} \sigma^2(x_i) = \sum_{i=1}^{n} \sigma^2(a_i) = \sum_{i=1}^{n} \beta_i$$

Faustregel: ca. 80 % der Gesamtvarianz sollten durch die ersten ℓ Hauptkomponenten erklärt werden.



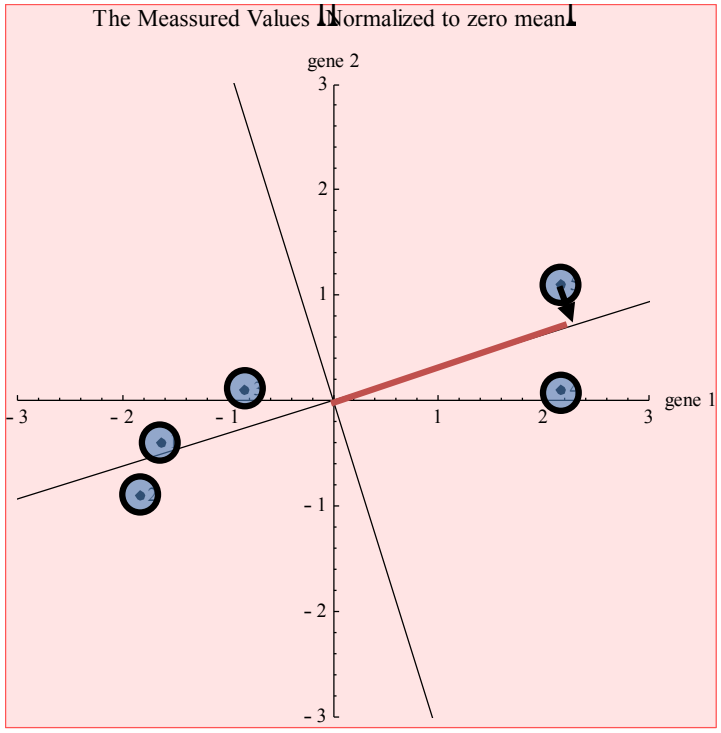
Gütekriterium der Approximation durch die ersten ℓ Hauptkomponenten:

$$P_k = \frac{\sum_{i=1}^{\ell} \sigma^2(a_i)}{\sigma^2_{\text{ges}}}$$

(erklärte Varianz)

Erklärte Varianz

Example Data Set 1



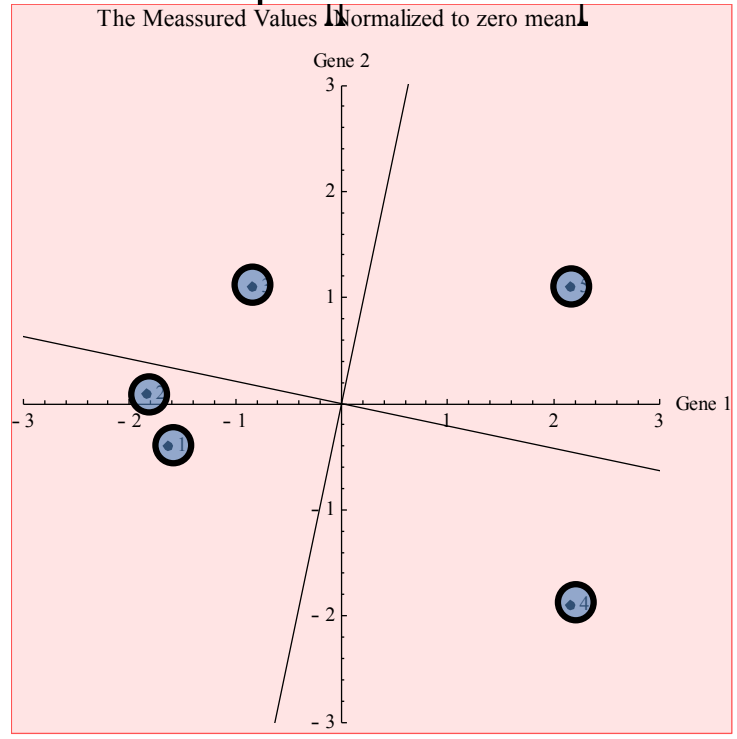
Variance:Sum of squares of all " ": —

{4.40469, 0.17331}

Explained variance percentage of total
96%=4.40 / (4.40+0.17), **4%.**

First component already
explains data to a great deal.

Example Data Set 2



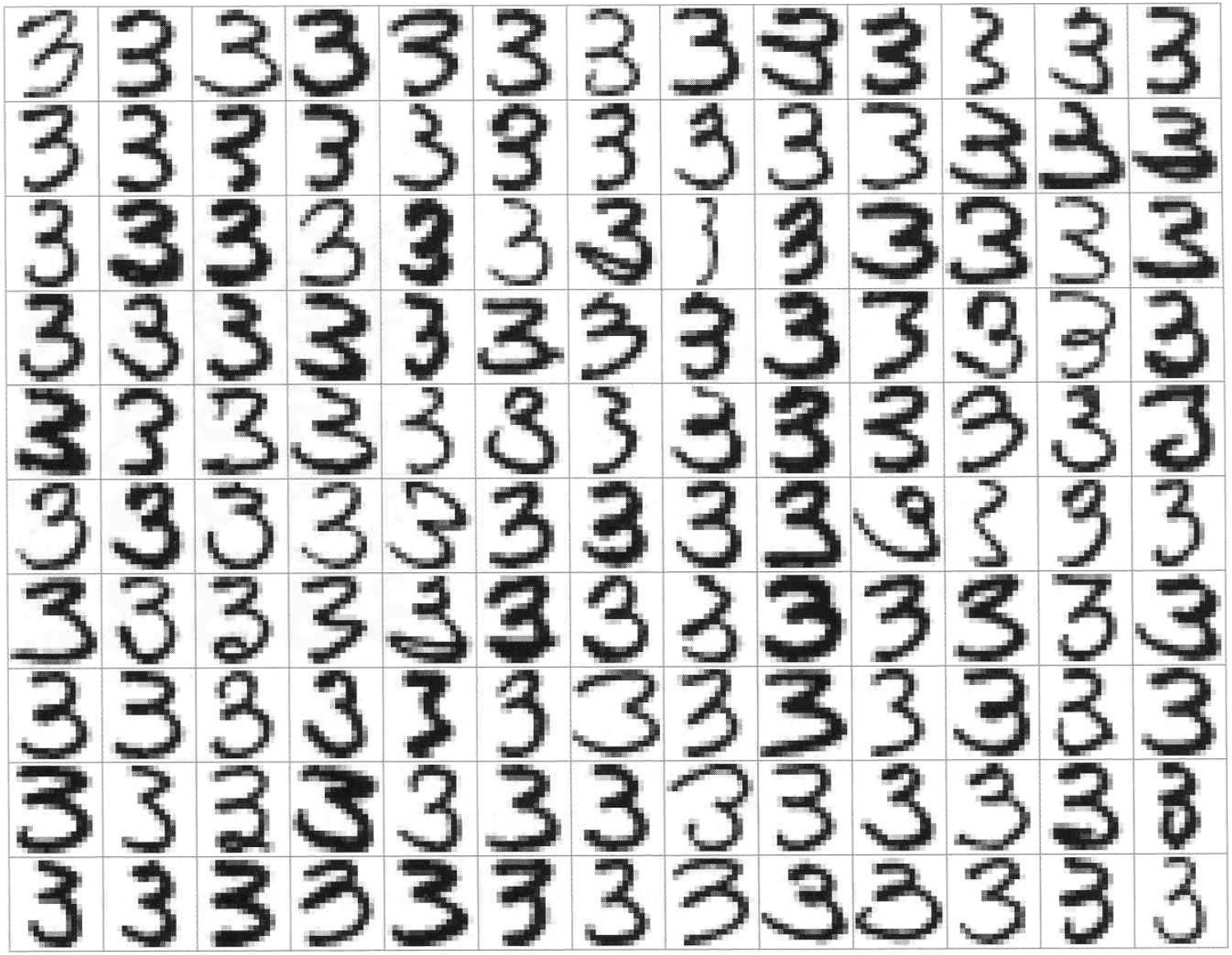
Variance:

{4.14257, 1.43543}

Explained variance:
74%, 26%.

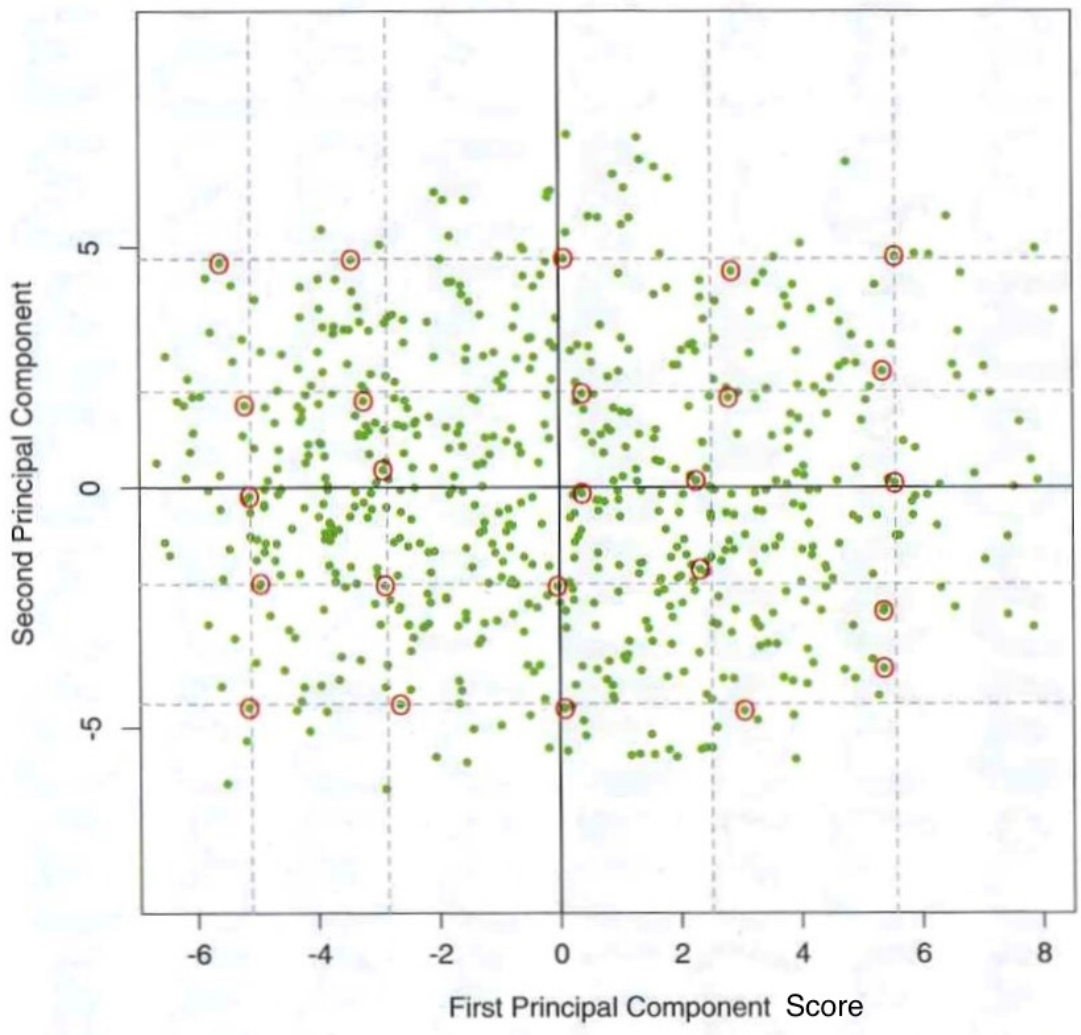
First component alone might
not be sufficient to explain the data.

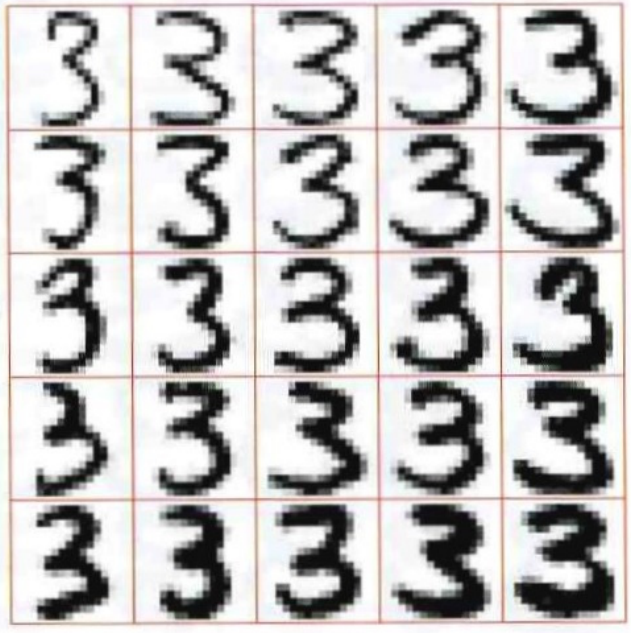
Beispiel: handgeschriebene Ziffern



[Haykin 99]

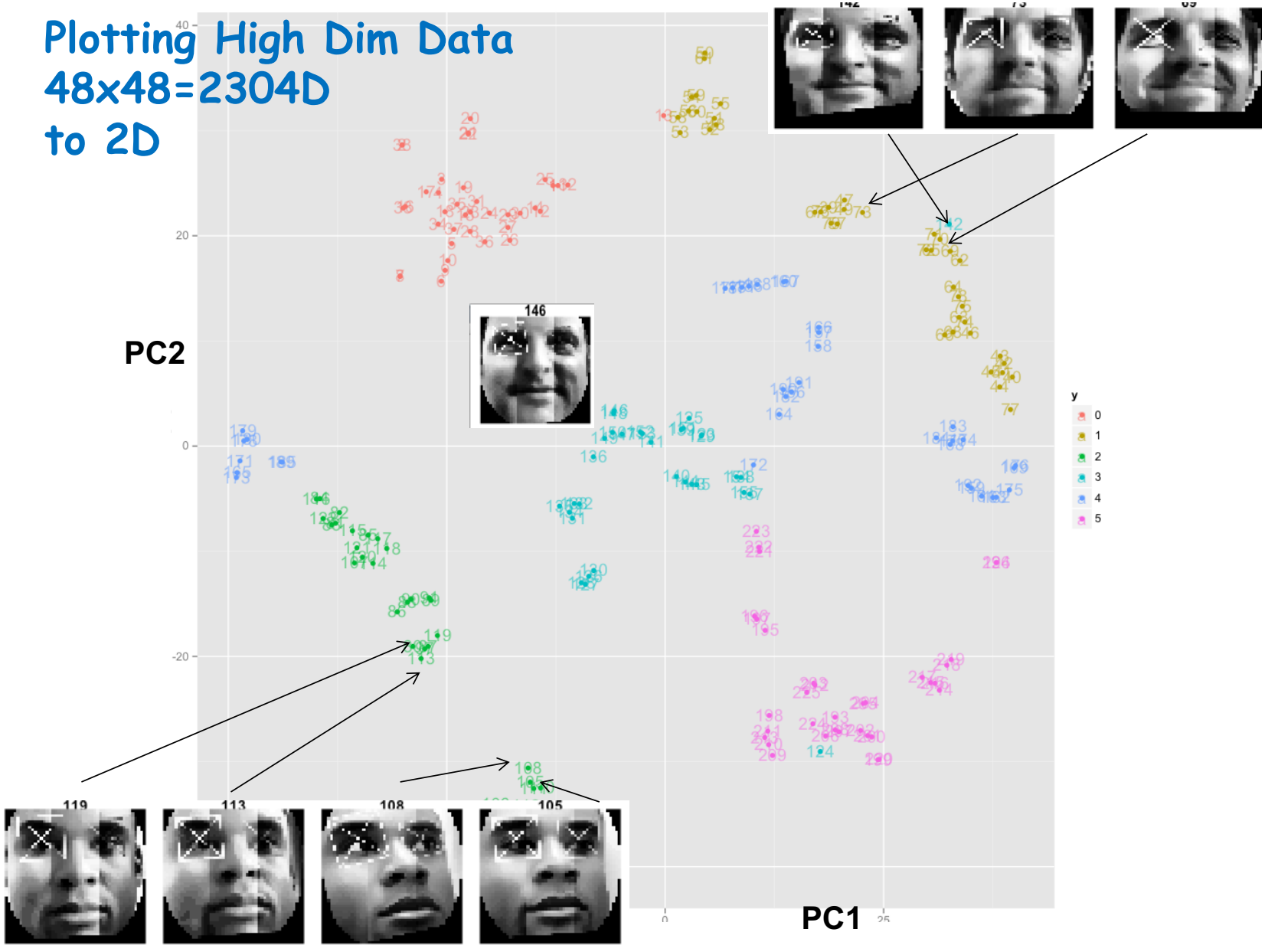
Erste 2 Hauptkomponenten der 3



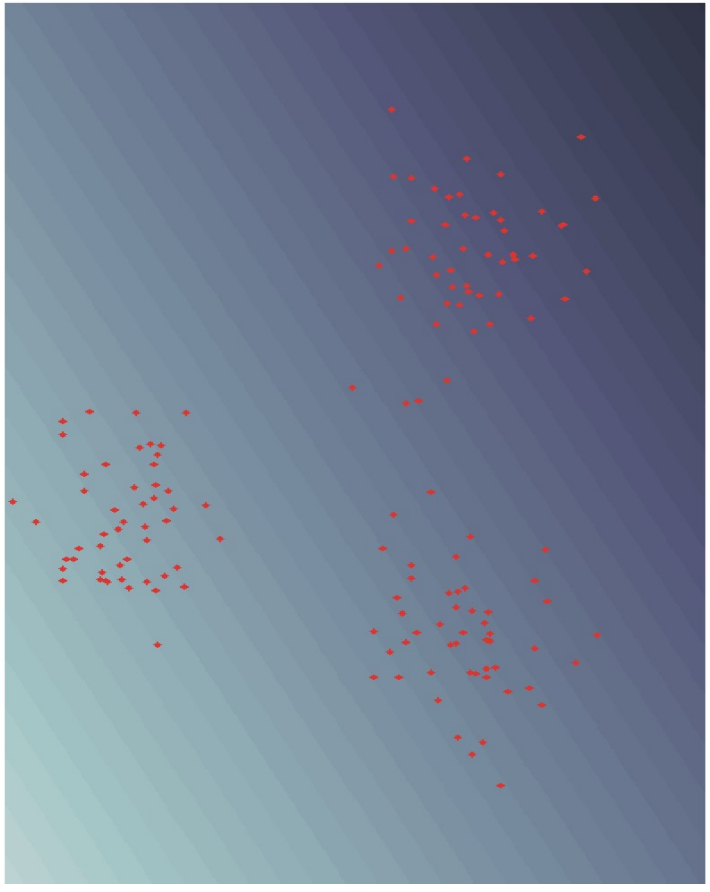


[Haykin 99]

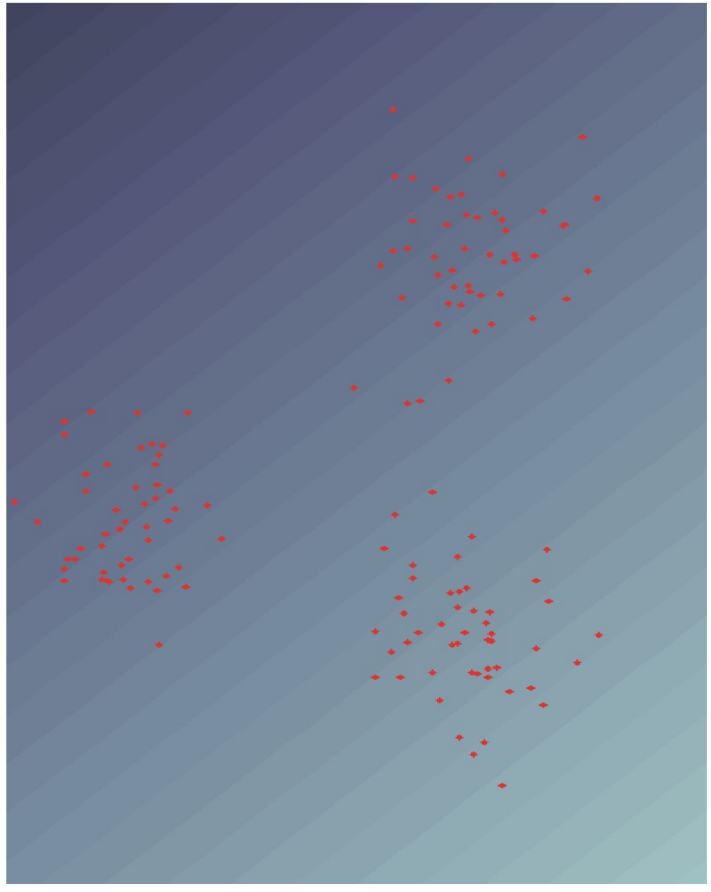
Beispiel: Gesichter



Hauptkomponentenanalyse bei 3 Clustern



1. Hauptkomponente



2. Hauptkomponente

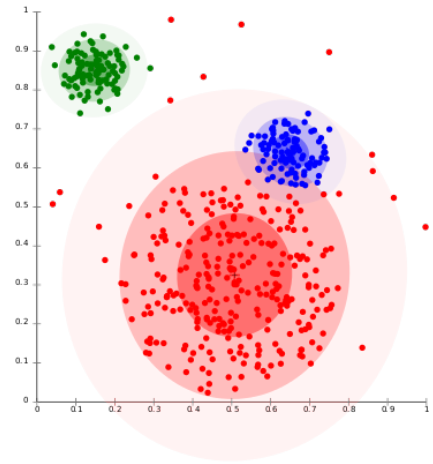
Übersicht

Clusteranalyse

Situation:

Die Daten werden in einem Merkmalsraum repräsentiert, ihrer Ähnlichkeit entspricht die Distanz im Merkmalsraum.

[Wikipedia]



Clusteranalyse:

geg. eine Objektmenge, ordne die Objekte in Gruppen von zueinander ähnlichen Mengen (Cluster), so dass Punkte im selben Cluster ähnlich sind, Punkte in verschiedenen Clustern weniger ähnlich.

- Die Clusteranalyse braucht ein Ähnlichkeits- oder Distanzmaß.
- Sie gehört zu den unüberwachten Lernverfahren, da die Daten dafür nicht gelabelt werden müssen.

Clusteranalyse ist nicht immer eindeutig

How many clusters?

Six Clusters

Two Clusters

Four Clusters

Ähnlichkeitsmaße und Gruppenbildung

Je nach Art der Variablen, die die Objekte beschreiben, braucht man unterschiedliche Ähnlichkeitsmaße:

- Binäre Variablen (nur zwei mögliche Werte): z.B. Jaccard-Koeffizient
- Nominale oder ordinale Variablen (mehrere diskrete Werte): z.B. Chi-Quadrat-Koeffizient, Phi-Koeffizient
- Metrische Variablen (kontinuierliche Wertebereiche): z.B. euklidische Distanz, Korrelation, Kovarianz, Korrelationskoeffizient

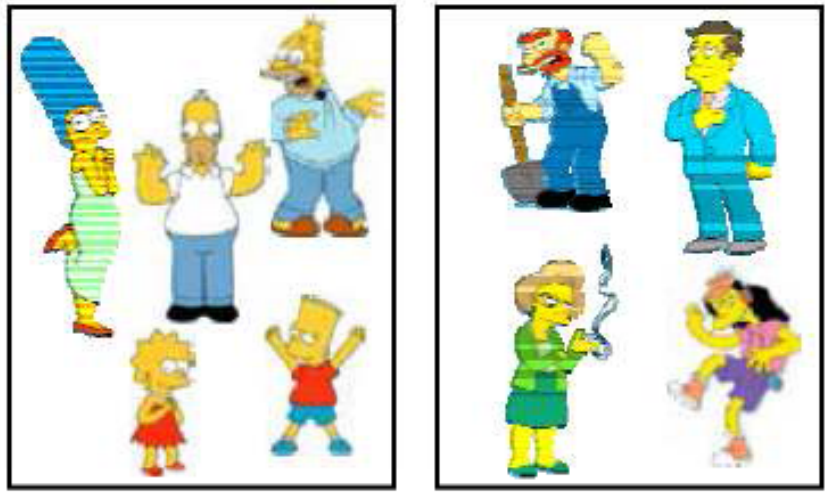
Man unterscheidet drei Formen der Gruppenbildung:

- Nichtüberlappend: Objekt kann nur in einem Cluster sein.
- Überlappend: ein Objekt kann in mehreren Clustern
- Fuzzy: jedes Objekt gehört einem Cluster mit einem bestimmten Grad des Zutreffens an.

Partitionierende und hierarchische Clusterverfahren

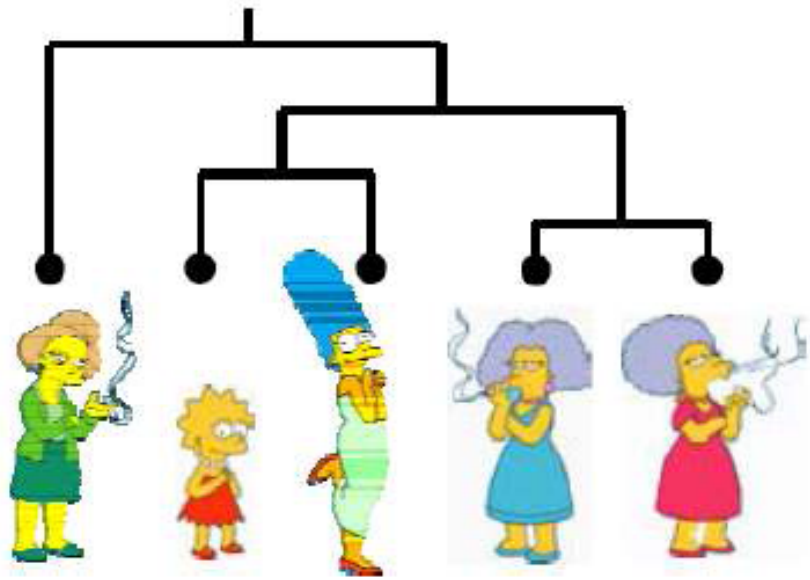
Partitionierende Clusterverfahren:

nichtüberlappende Gruppenbildung, Anzahl der Cluster wird festgelegt, die Partitionsgrenzen werden gelernt.



Hierarchische Clusterverfahren:

überlappende Gruppenbildung, fassen die Objekte hierarchisch zusammen, ausgehend von Einzelobejekten (agglomerativ, bottom up) oder von der Gesamtmenge (divisiv, top down).



Datenreduktion durch Clustern

- Durch Clustern kann eine große Menge von Daten durch eine kleinere Menge von Repräsentanten dargestellt werden. Dies erleichtert das Training vieler Lernalgorithmen, deren Laufzeit stark mit der Datenmenge ansteigt.
- Clustern kann auch zur Merkmalsextraktion benutzt werden: die durch einen Clusteralgorithmus gefundenen Cluster werden als **Codebuch** verwendet, in das neue Eingangsdaten projiziert werden. Die neuen Merkmale sind dann z.B. die Projektion oder die Distanz zu den Clusterzentren, oder einfach eine binäre Indikatorfunktion.
- Ein einfacher, häufig eingesetzter Clusteralgorithmus ist **k-Means** bzw. die Vektorquantisierung. Konvergiert beweisbar immer, ist aber NP-hart.
- Vorsicht: dieser Algorithmus funktioniert nur bei relativ niedrig-dimensionalen Daten gut, evtl. muss vorher eine Dimensionsreduktion erfolgen.

k-Means (Vektorquantisierung)

1. Initialisierung: geg. sind die Datenpunkte $\mathbf{x}_n$, $n = 1 \ldots N$, setze K Clusterzentren (means) auf Zufallswerte $\boldsymbol{\mu}_k^{(0)}$
2. Zuweisungsschritt: Weise jeden Datenpunkt $\mathbf{x}_n$ das nächstliegende Clusterzentrum $\hat{k}_n$ zu, d.h.

$$\hat{k}_n = \text{argmin}_k \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

   mit der Indikatorvariable $r_{kn} = 1$, falls $k = \hat{k}_n$, sonst $r_{kn} = 0$.
3. Anpassungsschritt: Berechne eine neue Position für das Clusterzentrum $\boldsymbol{\mu}_k$ aus den zugewiesenen Datenpunkten

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{kn}\mathbf{x}_n}{\sum_n r_{kn}}$$

4. Wiederhole Zuweisungs- und Anpassungsschritt, bis sich die Zuweisungen nicht mehr ändern.

2D-Beispiel für k-Means

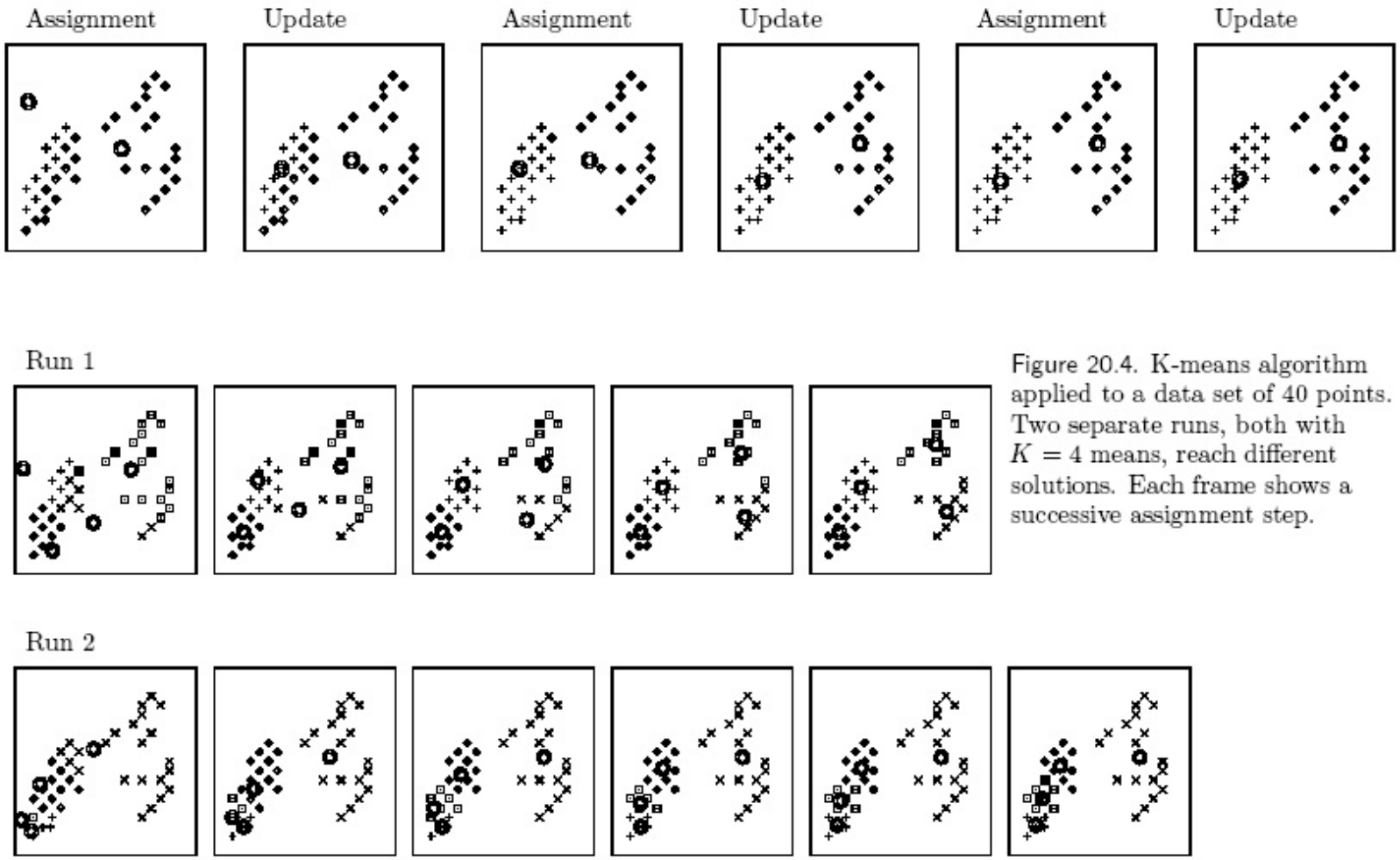


Figure 20.4. K-means algorithm applied to a data set of 40 points. Two separate runs, both with $K = 4$ means, reach different solutions. Each frame shows a successive assignment step.

[McKay 2003]

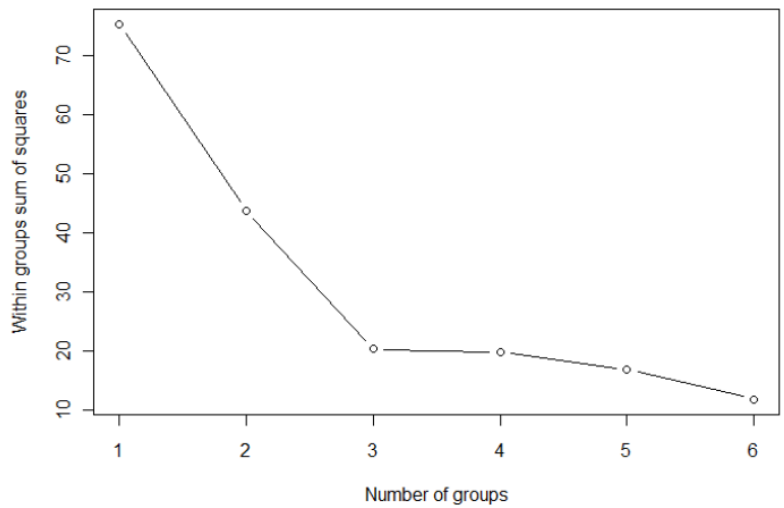
Was wird bei k-Means optimiert?

k-Means partitioniert die Datenpunkte so, dass die summierte Varianz innerhalb jedes Clusters minimiert wird:

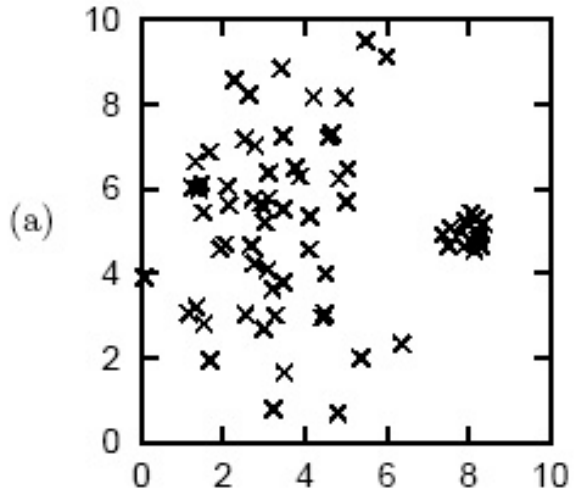
$$\text{minimiere}_{\boldsymbol{\mu}_k} \sum_{k=1}^{K} \frac{\sum_n r_{kn} (\mathbf{x}_n - \boldsymbol{\mu}_k)^2}{\sum_n r_{kn} - 1}$$

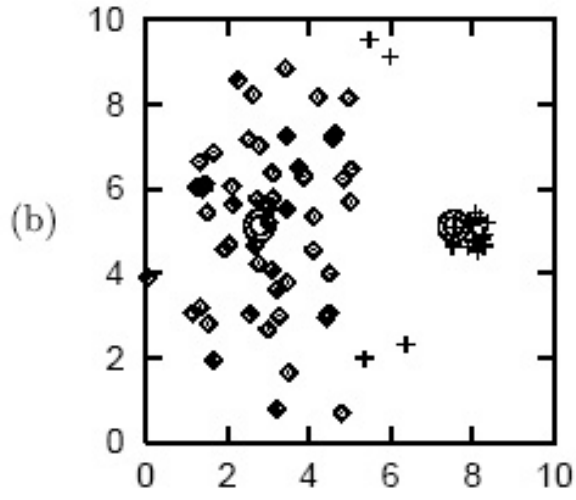
Leider kann der Algorithmus in lokalen Minima hängen bleiben, so dass ein Auffinden des globalen Minimums nicht garantiert ist. Das Kriterium kann dazu verwendet werden um die Anzahl der Cluster K festzulegen:

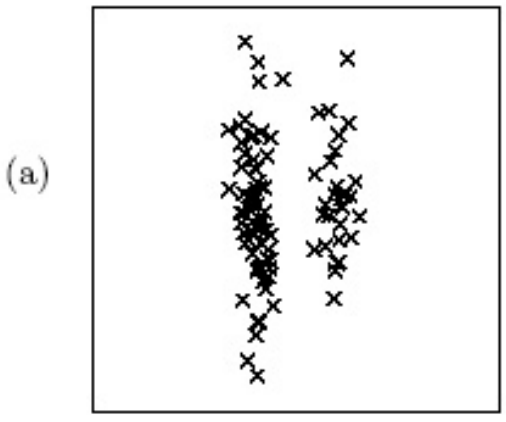
1. Algorithmus für unterschiedliche K durchlaufen lassen.
2. Summierte Varianz innerhalb jedes Clusters auftragen.
3. K nach dem letzten großen Abfall auswählen.

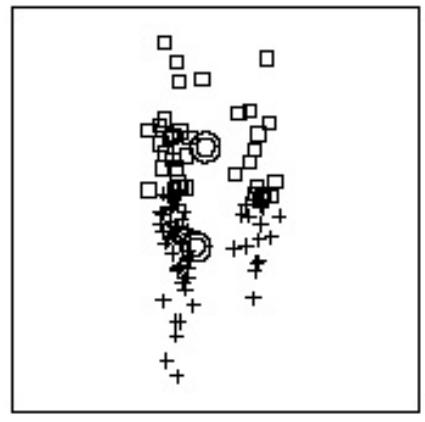


Problematische Fälle



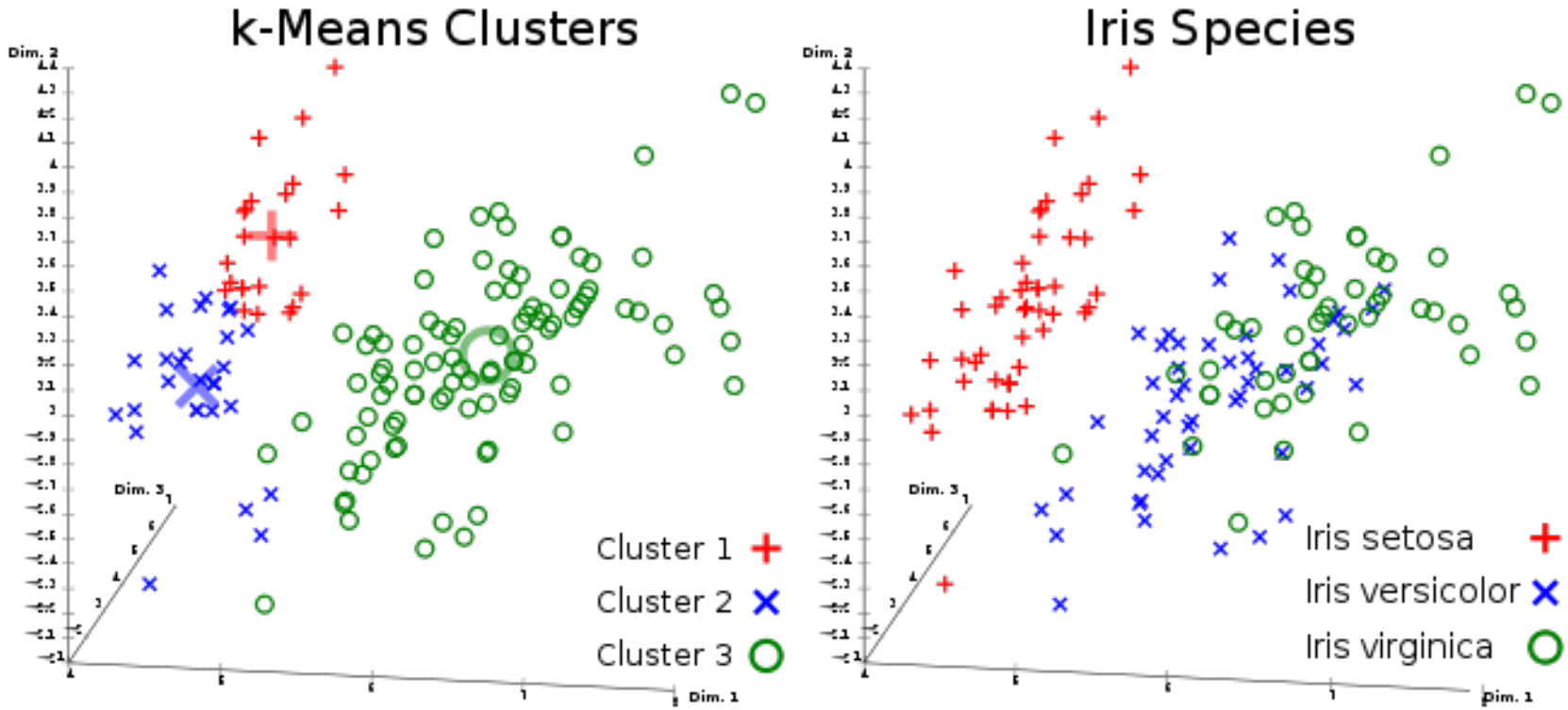






[McKay 2003]

Reale Daten: Iris-Datensatz



[Wikipedia]

Probleme von k-Means

Ad-hoc-Merkmale

- Die Anzahl der Cluster wird von vornherein festgelegt.
- Wie soll man sich entscheiden, wenn mehrere Clusterungen möglich sind?
- Ist der Mittelwert wirklich ein guter Repräsentant für einen Cluster?
- Gibt es bessere Distanzfunktionen?

Algorithmische Probleme

- Nur die Distanz zählt, Größe des Clusters wird nicht berücksichtigt.
- Keine Repräsentation der Form eines Clusters
- Harte Zuweisung: jeder Datenpunkt trägt nur zu einem Cluster bei, jeder Punkt hat innerhalb seines Clusters das gleiche Gewicht.